World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:2, 2009

Fast Codevector Search Algorithm for 3-D Vector Quantized Codebook

H. B. Kekre, and Tanuja K. Sarode

***Abstract*—**This paper presents a very simple and efficient algorithm for codebook search, which reduces a great deal of computation as compared to the full codebook search. The algorithm is based on sorting and centroid technique for search. The results table shows the effectiveness of the proposed algorithm in terms of computational complexity. In this paper we also introduce a new performance parameter named as Average fractional change in pixel value as we feel that it gives better understanding of the closeness of the image since it is related to the perception. This new performance parameter takes into consideration the average fractional change in each pixel value.

***Keywords*—**Vector Quantization, Data Compression, Encoding, Searching.

I. Introduction

VECTOR quantization (VQ) [1]-[3] is an efficient technique for data compression and has been successfully used in various applications involving VQ-based encoding and VQ-based recognition. The response time is very important factor for real time application [1]. Many type of VQ, such as classified VQ [37], [38], address VQ[37], [39], finite state VQ[37], [40], side match VQ[37], [41], mean-removed classified VQ[37], [42], and predictive classified VQ[37], [43], have been used for various purpose. VQ has been applied to some other applications, such as index compression [37], [44], and inverse half toning [37], [45], [46]. VQ has been very popular in a variety of research fields such as speech recognition and face detection [13], [47], pattern recognition [50]. VQ is also used in real time applications such as real time video-based event detection [13], [48] and anomaly intrusion detection systems [13], [49].

VQ can be defined as a mapping function that maps k-dimensional vector space to a finite set **CB = {C_1, C_2, C_3, ……, C_N}**. The set CB is called codebook consisting of N number of codevectors and each codevector $C_i = \{c_{i1}, c_{i2}, \ldots c_{ik}\}$ is of dimension k. The key to VQ is the good codebook.

H. B. Kekre is Senior Professor working with Computer Engineering Department, Mukesh Patel School of Technology Management and Engineering, NMIMS University, Vile Parle(w), Mumbai-56, INDIA. (phone: 9323557897; e-mail: hbkekre@yahoo.com).

Tanuja K. Sarode, is Ph.D. Scholar from Mukesh Patel School of Technology Management and Engineering, NMIMS University, Vile Parle(w), Mumbai-56, India and Assistant Professor working with Thadomal Shahani Engg. College, Bandra(W), Mumbai-50. (e-mail: tanuja_0123@yahoo.com).

Codebook can be generated in spatial domain by clustering algorithms or using transform domain techniques [6]-[8]. The method most commonly used to generate codebook is the Linde-Buzo-Gray (LBG) algorithm [3], [4] which is also called as Generalized Lloyd Algorithm (GLA).

In Encoding phase image is divided into non overlapping blocks and each block then converted to the training vector $\mathbf{X_i}$ **= (x_{i1}, x_{i2}, ……., x_{ik})**. The codebook is then searched for the nearest codevector C_{min} by computing squared Euclidean distance as presented in equation (1) with vector $\mathbf{X_i}$ with all the codevectors of the codebook **CB.** This method is called exhaustive search (ES).

$$d(X_i, C_{\min}) = \min_{1 \le j \le N}\{d(X_i, C_j)\}$$

$$\text{Where } d(X_i, C_j) = \sum_{p=1}^{k}(x_{ip}, c_{jp})^2 \qquad (1)$$

Although the Exhaustive Search (ES) method gives the optimal result at the end, it involves heavy computational complexity. If we observe the above equation (1) to obtain one nearest codevector for a training vector requires N Euclidean distance computation where N is the size of the codebook. So for M image training vectors, will require M*N number of Euclidean distances computations. It is obvious that if the codebook size is increased to reduce the distortion the searching time will also increase.

In order to reduce the searching time there are various search algorithms available in literature. So far, Partial Distortion search (PDS) [5], equal-average nearest neighbor search (ENNS) [9], the equal average equal variance nearest neighbor search (EENNS) [10], nearest neighbor search algorithm based on orthonormal transform (OTNNS) [11]. Partial Distortion Elimination (PDE) [25], triangular inequality elimination (TIE) [36-28], mean distance ordered partial codebook search (MPS) algorithm [20] ,double test algorithm (DTA) [22], fast codebook search algorithm based on the Cauchy-Schwarz inequality (CSI) [30], fast codebook search based on subvector technique (SVT) [31], the image encoding based on L_2-norm pyramid of codewords [32] and the fast algorithms using the modified L_2-norm pyramid (MLP) [33], fast codeword search algorithm based on MPS+TIE+PDE proposed by Yu-Chen, Bing-Hwang and Chih-Chiang (YBC) in 2008 [34], Eigen vector method (EVM) [21], and others [15], [19], [20], [22] are classified as partial search methods. Some of the partial techniques use data structure to organize the codebook for example tree-based [13], [14], [17], [18], [23], [35] and projection based structure [13], [16], [24]. All these algorithms reduce the computational cost needed for VQ encoding keeping the image quality equivalent to Exhaustive search algorithm.

In this paper we propose codebook search algorithm which uses sorting and centroid technique. The paper also compares the proposed algorithm with PDS, ENNS, EENNS, and OTNNS with respect to the execution time and the search efficiency in the form of ratio evaluated by how many times the Euclidean distance computation is averagely performed compared to the size of the codebook. A smaller ratio is better. We have also introduced a new performance parameter namely Average Fractional Change in Pixel Value (AFCPV) which is close to human perception. Smaller value of AFCPV refers to better performance.

In the next section we present some existing search algorithms. In section III proposed method is given followed by results in section IV and finally conclusions in section V.

II. Review of Different Search Algorithms

Some existing codevector search algorithms such as PDS, ENNS, EENNS, and OTNNS are reviewed in this section.

A. Partial Distortion Search (PDS)[5],[36]

The Partial distortion search (PDS) algorithm allows early termination of the distortion computation between input training vector and codevector by introducing a premature exit condition in the search algorithm. Let d_{min} be the smallest distortion obtained so far. If the codevector C_i satisfies the condition

$$\sum_{j=1}^{q}(x_{pj} - c_{ij}) \geq d_{\min} \qquad (2)$$

Where X_p is the image training vector and $j \leq q \leq k$ this guarantees that $d(x_p, c_i) \geq d_{min}$.

B. Equal average nearest neighbor search algorithm (ENNS)[9],[36]

The ENNS algorithm uses the fact that mean of the nearest codevector is usually close to the mean of the input vector. Let m_p and m_i be the mean values of training vector X_p and codevector C_i respectively. If the mean of the codevector C_i satisfies

$$m_i \geq m_p + \sqrt{d_{\min}/k} \text{ or } m_i \leq m_p - \sqrt{d_{\min}/k} \qquad (3)$$

then C_i will not be the nearest codevector to X_p. To perform ENNS algorithm mean of all the cedvectors should be computed off-line first and stored.

C. Equal Average Equal Variance Nearest Neighbor Search (EENNS)[9],[36]

EENNS algorithm introduces another significant feature of vector, the deviation, to reject codevectors. Let v_p, and v_i are the deviations of X_p and C_i respectively, then

$(v_p - v_i)^2 \leq d(X_p, C_i)$

If the deviation of the codevector C_i satisfies

$$v_i \geq v_p + \sqrt{d_{\min}} \text{ or } v_i \leq v_p - \sqrt{d_{\min}} \qquad (4)$$

then C_i will not be the nearest codevector to X_p.

EENNS algorithm performs in two steps. In the first step, if $m_i \geq m_p + \sqrt{d_{\min}/k}$ or $m_i \leq m_p - \sqrt{d_{\min}/k}$ then codevector C_i can be rejected. Otherwise, in the second, if

$$v_i \geq v_p + \sqrt{d_{\min}} \text{ or } v_i \leq v_p - \sqrt{d_{\min}} \qquad (5)$$

then codevector C_i can also be rejected. Hence this algorithms requires N mean values and N deviations of all codevectors.

D. Nearest Neighbor search algorithm based on orthonormal transform (OTNNS)[11]

Here orthonormal base vectors $V=(v_1, v_2, \ldots, v_k)$ for the Euclidean vector space R^k are considered. For any k-dimensional vector $x=(x_1, x_2, \ldots, x_k)$ it can be transformed to another Euclidean space defined by the k orthonormal base vectors, i.e. $x=\sum_{j=1}^{k} X_j v_j$ where $X(X_1, X_2, \ldots, X_k)$ is the coefficient vector in the transformed space.

In this algorithm each input vector is a 3-D residual vector and the orthonormal base vectors are $v_1 = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$, $v_2 = (1/\sqrt{6}, 1/\sqrt{6}, -2/\sqrt{6})$, $v_3 = (1/\sqrt{2}, -1/\sqrt{2}, 0)$

The conditions for judging possible nearest codevectors are

$$X_{i,min} \leq Y_{ji} \leq X_{i,max} \quad 1 \leq i \leq 3 \qquad (6)$$

Where $Y_j = (Y_{j1}, Y_{j2}, Y_{j3})$ is the coefficient vector of C_j in the transformed space, and

$$X_{i,min} = X_i - d_{min} \qquad (7)$$

$$X_{i,max} = X_i - d_{min} \qquad (8)$$

Preprocessing:

Transform each codevector of the codebook into the space with base vector $V=(v_1, v_2, v_3)$ and then sort codevectors in ascending order with respect to the first elements, i.e. the coefficients along the base vector v_1.

Online step:

For searching each input vector X_i is transformed to obtained $\hat{X}_i$. The probable nearby codevector Y_j is gussed based on the minimum first element difference criterion. d_{min}, $X_{i,min}$, $X_{i,max}$ are calculated. For each codevector Y_j first check if (6) is satisfied. If not then Y_j is rejected else $d(\hat{X}_i, Y_j)$ is calculated. If $d(\hat{X}_i, Y_j) < d_{min}$, then the current closest codevector to $\hat{X}_i$ is taken as Y_j with d_{min} set to $d(\hat{X}_i, Y_j)$ and $X_{i,min}$ and $X_{i,max}$ are updated accordingly. The procedure is repeated until best match is found.

III. Proposed Algorithm

Let CB be the codebook consisting of k-dimensional vectors. In this paper k=3 consisting of R, G, B component values of every pixel. First sort the codebook with respect to the first element of the codevector and then compute the centroid c_0 of the first elements of all the codevectors. The codebook is then divided into two parts based on the centroid of the first element, the upper part consists of element values less than this centroid. The upper part of the codebook is again sorted with respect to the second elements of the codevectors and again centroid c_{00} is computed for the second element for the upper part. The process is repeated for the lower part too i.e. lower part of codebook is also sorted with respect to the second elements of the codevectors corresponding the lower part of the codebook and centroid c_{01} is computed for the lower part. Based on the centroid the upper part of the codebook is further divided in to two parts and the above process is repeated, similarly lower part of the codebook is also divided based on to centroid and above process is repeated. For codebook of size N the above process is repeated for $r=(\log_2 N - 3)$ times so get 2^r parts of the codebook. The formation of the codebook into subparts is a preprocessing step for encoding is depicted in Fig. 1 as shown below.

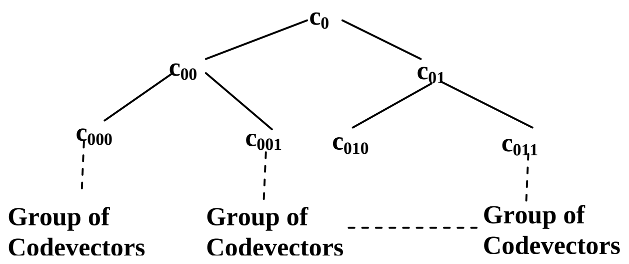

Fig. 1 Dividing codebook into subparts

Online process:

In Encoding step the first element x_{i1} of image training vector X_i is compared with the c_0, if $x_{i1} < c_0$ then x_{i2} is compared with c_{00} else then x_{i2} is compared with c_{01} and so on. Once the training vector reaches the last level of tree the nearest codevector is searched form the group of codevectors using Euclidean distance computation. Instead of full search we are dividing codebook into subparts, nearest subpart for the training vector is found out and then closest codevector is searched using exhaustive search applied only to the subpart that is obtained. To locate the nearest subpart $(\log_2 N - 3)$ comparisons are required.

IV. Results

The proposed algorithm is compared with PDS, ENNS, EENNS and OTNNS experimented on 3D mesh model obtained from famous Princeton 3D mesh library [12], Standford Bunny and Standford Dragon. The algorithms are implemented using Pentium IV 1.7 GHz 512 MB RAM, using Matlab 6. The algorithm requires 2^r-1 extra memory space to store the centroids.

Here we have introduced a new performance parameter which is named as Average Fractional Change in Pixel Value (AFCPV) and it is computed as follows:

$$\sum_{MN} \frac{\left| f(x,y) - \hat{f}(x,y) \right|}{f(x,y)} \tag{9}$$

Where $f(x,y)$ is original image of size MxN and $\hat{f}(x,y)$ is the reconstructed image,

We feel that our new performance parameter AFCPV gives better understanding of the closeness of the image as it is related to the perception. It takes into consideration the average fractional change in each pixel value. To void division by zero problem we have replaces zero by one in the denominator in equation (9). However it has been observed that there were not more than ten pixels having zero value in the original image.

Table I shows the time needed for encoding Bunny and Dragon images for codebooks of sizes 256 and 1024. The distortion of the encoded image is also shown for the codebooks 256 and 1024. The distortion of the encoded image remains same for all the algorithms since they are full-search equivalent.

Table II shows the search efficiency in the form of ratio evaluated by how many times the Euclidean distance computation is averagely performed compared to the size of the codebook. Search space for ES is considered as 100% and the reduced search space for other algorithms are compared to ES. A smaller ratio is better.

Table III Shows PSNR, Time, search efficiency and AFCPV for the proposed algorithm for different codebooks sizes.

Fig. 2 shows the results for Bunny image using codebook of sizes 256, 512, 1024 and 2048 encoded using proposed search algorithm.

TABLE I
COMPARISON OF SEARCH ALGORITHMS WITH RESPECT TO EXECUTION TIME

Images	CB Size	PSNR	FS	PDS	ENNS	EENNS	OTNNS	Proposed
Bunny	256	49.9	0.49	0.3	0.08	0.09	0.04	0.0041
	1024	54.2	1.94	1.02	0.16	0.14	0.07	0.0041
Dragon	256	49.9	1.45	0.86	0.25	0.28	0.15	0.0041
	1024	53.5	5.34	2.89	0.44	0.41	0.2	0.0042

TABLE II
RATIO OF REDUCED SEARCH SPACE AFTER EACH CHECK COMPARED TO ES

Images	CB Size	PDS	ENNS	EENNS	OTNNS	Propo-sed
Bunny	256	11.26	7.2	2.79	1.5	0.1
	1024	3.59	3.19	0.84	0.4	0.02
Dragon	256	11.9	7.6	3	1.52	0.1
	1024	3.67	3.65	1	0.43	0.02

TABLE III
RESULT OF THE PROPOSED ALGORITHM FOR DIFFERENT CODEBOOKS SIZES

	Proposed Algorithm				
Images	**CB Size**	**PSNR**	**Time**	**Search Efficiency**	**AFCPV**
Bunny	256	49.9	0.0041	0.10	0.6483
	512	52.3	0.0042	0.05	0.485
	1024	54.2	0.0041	0.02	0.345
	2048	55.9	0.004	0.01	0.2488
Dragon	256	49.9	0.0042	0.10	0.4901
	512	51.8	0.0041	0.05	0.3679
	1024	53.5	0.0042	0.02	0.2765
	2048	55.0	0.0043	0.01	0.2372

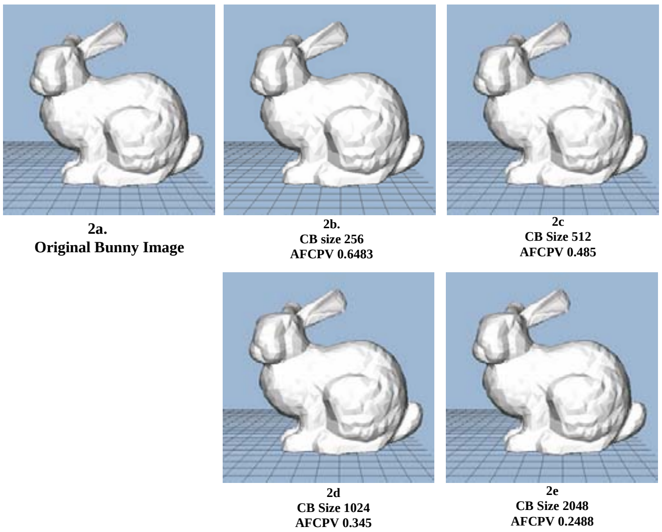

Fig. 2 Results of Bunny image using proposed search algorithm for the codebooks of different sizes 256, 512, 1024 and 2048

V. CONCLUSION

From Table I and Table II it is observed that proposed algorithm is faster as compared to other search algorithms since it requires considerably less number of Euclidean distance computations. Table III gives the performance of our proposed algorithm for different codebook sizes. The search efficiency in the form of ratio evaluated by how many times the Euclidean distance computation is averagely performed compared to the size of the codebook. Search space for ES is considered as 100% and the reduced search space for other algorithms are compared to ES. A smaller ratio is better. It is observed that proposed algorithm gives better search efficiency as compared to other algorithms. The newly introduced performance parameter AFCPV is computed for the proposed algorithm for different codebook sizes it is observed that lager the codebook size gives lower value of AFCPV representing better human perception.

REFERENCES

[1] Jeng-Shyang Pan, Zhe-Ming Lu, and Sheng-He Sun.: ‘An Efficient Encoding Algorithm for Vector Quantization Based on Subvector Technique’, *IEEE Transactions on image processing*, vol 12 No. 3 March 2003.

[2] R. M. Gray.: ‘Vector quantization’, *IEEE ASSP Marg*, Apr. 1984, pp. 4-29.

[3] Y. Linde, A. Buzo, and R. M. Gray.: ‘An algorithm for vector quantizer design,” *IEEE Trans. Commun*.’, vol. COM-28, no. 1, 1980, pp. 84-95.

[4] A. Gersho, R.M. Gray.: ‘*Vector Quantization and Signal Compressio’*, Kluwer Academic Publishers, Boston, MA, 1991.

[5] C. D. Bei and R. M. Gray.: ‘An improvement of the minimum distortion encoding algorithm for vector quantization’, *IEEE Trans. Commun.*,vol. 33, no. 10, pp. 1132–1133, Oct. 1985.

[6] Momotaz Begum, Nurun Nahar, Kaneez Fatimah, M. K. Hasan, and M. A. Rahaman: ‘An Efficient Algorithm for Codebook Design in Transform Vector Quantization’, *WSCG’2003*, February 3-7, 2003.

[7] Robert Li and Jung Kim: ‘Image Compression Using Fast Transformed Vector Quantization’, *IEEE Applied Imagery Pattern Recognition Workshop, 2000 Proceedings* 29th Volume , Issue , 2000 Page(s):141 – 145.

[8] Zhibin Pan; Kotani, K.; Ohmi, T., ‘Enhanced fast encoding method for vector quantization by finding an optimally-ordered Walsh transform kernel’**,** *ICIP 2005, IEEE International Conference*, Volume 1, Issue, 11-14, Page(s): I - 573-6, Sept. 2005.

[9] Guan, L., and Kamel, M. : ‘Equal-average hyperplane partitioning method for vector quantization of image data’, Patt. Recognit. Lett., 1992, pp. 693-699.

[10] Lee, H., and Chen, L. H. : ‘Fast closest codevector search algorithms for vector quantization’, Signal Process., 1995, 43, pp. 323-331.

[11] Z. Li, and Z.- M. Lu. : ‘Fast codevector search scheme for 3D mesh model vector quantization’, Electron. Lett., 2008, 44, pp. 104-105.

[12] Princeton University, “3D model search engine”, http://shape.cs.princeton.edu

[13] Chin-Chen Chang, Wen-Chuan Wu, “ Fast Planar-Oriented Ripple Search Algorithm for Hyperspace VQ Codebook*”, IEEE Transaction on image processing,* vol 16, no. 6, June 2007.

[14] C. C. Chang and T. S. Chen, “New tree-structured vector quantization with closest-coupled multipath searching method,” *Opt. Eng.*, vol. 36, no. 6, pp. 1713–1720, Jun. 1997.

[15] C. C. Chang and I. C. Lin, “Fast search algorithm for vector quantization without extra look-up table using declustered subcodebooks,” *IEE Proc. Vis., Image, Signal Process.*, vol. 152, no. 5, pp. 513–519, Oct.2005.

[16] C. C. Chang, D. C. Lin, and T. S. Chen, “An improved VQ codebook search algorithm using principal component analysis,” *J. Vis. Commun. Image Represent.*, vol. 8, no. 1, pp. 27–37, Mar. 1997.

[17] C. C. Chang, F. J. Shiue, and T. S. Chen, “Tree structured vector quantization with dynamic path search,” in *Proc. Int. Workshop on Multimedia Network Systems*, Aizu, Japan, pp. 536 –541, Sep. 1999.

[18] R. M. Gray and Y. Linde, “Vector quantization and predictive quantizers for gauss-markov sources,” *IEEE Trans. Commun.*, vol. 30, no. 2, pp. 381–389, Feb. 1982.

[19] C. M. Huang, Q. Bi, G. S. Stiles, and R. W. Harris, “Fast full-search equivalent encoding algorithms for image compression using vector quantization,” *IEEE Trans. Image Process.*, vol. 1, no. 3, pp. 413–416, Jul. 1992.

[20] Y. C. Hu and C. C. Chang, “An effective codebook search algorithm for vector quantization”, *Imag. Sci. J.*, vol. 51, no. 4, pp. 221–234, Dec. 2003.

[21] C. H. Lee and L. H. Chen, “High-speed closest codeword search algorithm for vector quantization,” *Signal Process.*, vol. 43, no. 3, pp.323–331, May 1995.

[22] L. Torres and J. Huguet, “An improvement on codebook search for vector quantisation”, *IEEE Trans. Commun.*, vol. 42, no. 2, pp. 208–210, Feb. 1994.

[23] S. J. Wang and C. H. Yang, “Hierarchy-oriented searching algorithms using alternative duplicate codewords for vector quantization mechanism,” *Appl. Math. Comput.*, vol. 162, no. 234, pp. 559–576, Mar. 2005.

[24] S. C. Tai, C. C. Lai, and Y. C. Lin, “Two fast nearest neighbor searching algorithms for image vector quantization,” *IEEE Trans. Commun.*, vol. 44, no. 12, pp. 1623–1628, Dec. 1996.

[25] C. Bei, R.M. Gray, ''An improvement of the minimum distortion encoding algorithm for vector quantization'', IEEE Trans. Commun.33, 1985, pp. 1132–1133.
[26] S.H. Huang, S.H. Chen, ''Fast encoding algorithm for VQ-based image coding'', Electron. Lett. Vol. 26, issue 19, 1990, pp. 1618–1619.
[27] W. Li, E. Salari, ''A fast vector quantization encoding method for image compression'', IEEE Trans. Circ. Syst. Vid. Vol 5, 1995, pp. 119–123.
[28] C.H. Hsieh, Y.J. Liu, ''Fast search algorithms for vector quantization of images using multiple triangle inequalities and wavelet transform'', IEEE Trans. Image Process. Vol. 9, issue 3, 2000, pp. 321–328.
[29] S.W. Ra, J.K. Kim, ''A fast mean-distance-ordered partial codebook search algorithm for image vector quantization'', IEEE Trans. Circuits-II, vol. 40, issue 9, 1993, pp. 576–579.
[30] K.S. Wu, J.C. Lin, ''Fast VQ encoding by an efficient kick-out condition'', IEEE Trans. Circ. Syst. Vid., vol.10, issue 1, 2000, pp. 59–62.
[31] J.S. Pan, Z.M. Lu, S.H. Sun, ''An efficient encoding algorithm for vector quantization based on subvector technique'', IEEE Trans. Image Process. Vol 12, issue 3, 2003, pp. 265–270.
[32] B.C. Song, J.B. Ra, ''A fast algorithm for vector quantization using L2-norm pyramid of codewords'', IEEE Trans. Image Process. Vol. 4, issue 12, 2002, pp. 325–327.
[33] Z. Pan, K. Kotani, T. Ohmi, ''Fast encoding method for vector quantization using modified L2-norm pyramid'', IEEE Signal Process. Lett. Vol. 12, issue 9, 2005, pp. 609–612.
[34] Y. Chen, B. Hwang, C. Chiang, "Fast VQ codebook search algorithm for grayscale image coding", Image and Vision Compu. 26, 2008, pp. 657-666.
[35] Dr. H. B. Kekre, Ms. Tanuja K. Sarode, "Binary Tree Based Fast Search Algorithm for Closest Codevector in the Codebook for Vector Quantization", SPIT-IEEE Colloquium 2007, SPCE, Mumbai, 4th – 5th February 2007.
[36] Shu-Chuan Chu, Zhe-Ming Lu, Jeng-Shyang Pan, "Hadamard transform based fast codeword search algorithm for high-dimensional VQ encoding", Information Sciences, 177, 2007, pp. 734-746.
[37] Jim Z.C. Lai, Yi-Ching Liaw and Julie Liu, "A fast VQ codebook generation algorithm using codeword displacement" , *Pattern Recogn.* vol. 41, no. 1, pp 315–319, 2008.
[38] Y.C. Liaw, J.Z.C. Lai, W. Lo, Image restoration of compressed image using classified vector quantization, *Pattern Recogn*. vol. 35, No. 2, pp 181–192, 2002.
[39] N.M. Nasrabadi, Y. Feng, Image compression using address vector quantization, *IEEE Trans. Commun*. vol. 38 No. 12, pp. 2166–2173, 1990.
[40] J. Foster, R.M. Gray, M.O. Dunham, Finite state vector quantization for waveform coding, *IEEE Trans. Inf. Theory* vol. 31, No. 3, pp. 348–359, 1985.
[41] T. Kim, Side match and overlap match vector quantizers for images, *IEEE Trans. Image Process*. vol. 1, No. 2, pp. 170–185, 1992.
[42] J.Z.C. Lai, Y.C. Liaw, W. Lo, Artifact reduction of JPEG coded images using mean-removed classified vector quantization, *Signal Process*. vol. 82, No. 10, pp. 1375–1388, 2002.
[43] K.N. Ngan, H.C. Koh, Predictive classified vector quantization, *IEEE Trans. Image Process.* vol. 1, No. 3, pp. 269–280, 1992.
[44] C.H. Hsieh, J.C. Tsai, Lossless compression of VQ index with search order coding, *IEEE Trans. Image Process*. vol. 5, No. 11, pp. 1579–1582, 1996.
[45] J.Z.C. Lai, J.Y. Yen, Inverse error-diffusion using classified vector quantization, *IEEE Trans. Image Process.* vol. 7, No. 12, pp. 1753–1758, 1998.
[46] P.C. Chang, C.S. Yu, T.H. Lee, "Hybrid LMS-MMSE inverse halftoning technique", *IEEE Trans. Image Process*. vol. 10, No. 1, pp. 95–103, 2001.
[47] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Trans. Multimedia*, vol. 1, no. 3, pp. 264–277, Sep. 1999.
[48] H. Y. M. Liao, D. Y. Chen, C. W. Su, and H. R. Tyan, "Real-time event detection and its applications to surveillance systems," in *Proc. IEEE Int. Symp. Circuits and Systems*, Kos, Greece, pp. 509–512, May 2006.
[49] J. Zheng and M. Hu, "An anomaly intrusion detection system based on vector quantization," *IEICE Trans. Inf. Syst.*, vol. E89-D, no. 1, pp. 201–210, Jan. 2006.
[50] Ahmed A. Abdelwahab, Nora S. Muharram, "A Fast Codebook Design Algorithm Based on a Fuzzy Clustering Methodology", *International Journal of Image and Graphics*, vol. 7, no. 2 pp. 291-302, 2007.

H. B. Kekre has received B.E. (Hons.) in Telecomm. Engineering. from Jabalpur University in 1958, M.Tech (Industrial Electronics) from IIT Bombay in 1960, M.S.Engg. (Electrical Engg.) from University of Ottawa in 1965 and Ph.D. (System Identification) from IIT Bombay in 1970. He has worked Over 35 years as Faculty of Electrical Engg. and then HOD Computer Science and Engg. at IIT Bombay. For last 13 years worked as a Professor in Department of Computer Engg. at TSEC, Mumbai. He is currently Senior Professor working with Mukesh Patel School of Technology Management and Engineering, NMIMS University, Vile Parle(w), Mumbai, INDIA. His areas of interest are Digital Signal processing, Image Processing and Computer Networks. He has more than 200 papers in National / International Conferences / Journals to his credit. Recently five students working under his guidance have received best paper awards.

Tanuja K. Sarode has Received M.E.(Computer Engineering) degree from Mumbai University in 2004, currently Perusing Ph.D. from NMIMS University, Mumbai. She has more than 9 years of experience in teaching. Currently working as a Assistant Professor in Department of Computer Engineering at Thadomal Shahani Engineering College, Mumbai. Her areas of interest are Image Processing, Signal Processing and Computer Graphics. She has 15 papers in National /International Conferences/journal to her credit.